Solving First-Order Constraints in the Theory of the Evaluated Trees

Thi-Bich-Hanh Dao, Khalil Djelloul

Solving First-Order Constraints in the Theory of the Evaluated Trees

Thi-Bich-Hanh Dao[1] and Khalil Djelloul[2]

[1] LIFO, Bat. 3IA, rue Léonard de Vinci. 45067 Orléans, France.
[2] LIF, Parc scientifique et technologique de Luminy. 13288 Marseille, France.

Abstract. We describe in this paper a general algorithm for solving first-order constraints in the theory T of the evaluated trees which is a combination of the theory of finite or infinite trees and the theory of the rational numbers with addition, subtraction and a linear dense order relation. It transforms a first-order formula φ, which can possibly contain free variables, into a disjunction ϕ of solved formulas which is equivalent in T, without new free variables and such that ϕ is either *true* or *false* or a formula having at least one free variable and being equivalent neither to *true* nor to *false* in T.

1 Introduction

The theory of finite or infinite trees plays a fundamental role in computer science. There exists algorithms for eliminating quantifiers which can decide the validity of propositions in these theories [7, 3]. We have extended this theory by giving a complete first-order axiomatization T of the evaluated trees [6] which are combination of finite or infinite trees with construction operations and the rational numbers with addition, subtraction and a linear dense order relation. This theory reflects essentially to Prolog III and IV which have been modeled using combination of trees, rational numbers, booleans and intervals [2, 1]. In this paper we describe a general algorithm for solving first-order constraints in T, i.e. in all models of T. Our aim is not only to decide the validity of the propositions, but to be able to express solutions of constraints in T, which can possibly contain free variables, in a simple and explicit way.

2 Description of the algorithm

The algorithm is not simply a combination of an algorithm over trees with one over rational numbers, but a powerful mechanism which is able to solve any first-order constraint containing typed/untyped variables and presents the solutions on the free variables in a clear and explicit way. One of the major difficulty resides in the fact that the theory of trees does not accept full elimination of quantifiers and the function symbols $+$ and $-$ of T have two different behaviors whether they are applied on trees or on rational numbers. The main points of the algorithm are (full description of the algorithm can be found in [5]):

- We define basic formulas, which are conjunctions of formulas of the forms *true*, *false*, $num\, x$, $tree\, x$, $x = y$, $x = fy_1...y_n$, $\Sigma_{i=1}^{n} a_i x_i = a_0 1$, $\Sigma_{i=1}^{n} a_i x_i < a_0 1$, with x_i, y_j variables and $a_i \in \boldsymbol{Z}$ and give a mechanism to derive typing constraints in basic formulas.
- We define blocks, which are basic formulas where all the variables are typed and where there is no type conflict and define solved blocks.
- We give the definition of working formulas, which are formulas written by using only existentially quantified blocks and their negation. To be able to control the rewriting rules, which are the heart of the algorithm, we give to each negation symbol $\neg$ a number, which will present properties of the corresponding working formula. Using these number, we define initial, final working formulas, and solving formulas.
- We give a set of 28 rewriting rules, which transform an initial working formula into a conjunction of final working formulas, which can be directly transformed into a disjunction of solving formulas, which is either *true* or *false* or a formula φ having at least one free variable such that neither $T \models \varphi$ nor $T \models \neg\varphi$. These rules make top-down local solvings and propagations and bottom-up quantifier eliminations and distributions. We show that these rules are correct and terminating.

3 Conclusion

The non-elementarity of the decision in the theory of trees is known [8], i.e the complexity of the decision cannot be bounded by a tower of powers of 2's with a fixed height. Thus our algorithm must not escape this kind of complexity in the worst case. We have programmed a similar algorithm on the theory of trees and nevertheless could be able to solve formulas of winning positions in two partner games, having 160 nested quantifiers [4]. Actually we try to extend our algorithm to any combination of a theory T with the theory of finite or infinite trees.

References

1. Benhamou F, and al. Le manuel de Prolog IV , PrologIA, Marseille, France, 1996.
2. Colmerauer A. An introduction to Prolog III. Comm. of the ACM, 33(7):68–90,1990.
3. Comon H. Résolution de contraintes dans des algèbres de termes. Rapport d'Habilitation, Université de Paris Sud, 1992.
4. Dao T. Résolution de contraintes du premier ordre dans la théorie des arbres finis ou infinis. Thèse d'informatique, Université de la Méditerranée, 2000.
5. Dao TBH, Djelloul K., Solving First-Order Constraints in the Theory of the Evaluated Trees, Rapport de recherche LIFO RR-2006-05, `http://www.univ-orleans.fr/lifo/rapports.php`
6. Djelloul K. About the combination of trees and rational numbers in a complete first-order theory. 5th Int. Conf. FroCoS 2005, LNAI, vol 3717, P. 106–122.
7. Maher M. Complete axiomatization of the algebra of finite, rational and infinite trees. *Technical report, IBM - T.J.Watson Research Center*, 1988.
8. Vorobyov S. An Improved Lower Bound for the Elementary Theories of Trees, CADE'96, LNAI 1104, pp. 275- 287, 1996.